Docket No.: 20910/0206207-US0
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Letters Patent of:
Cynthia A. McGuire et al.

Patent No.: 7,328,376

Issued: February 5, 2008

For: ERROR REPORTING TO DIAGNOSTIC ENGINES BASED ON THEIR DIAGNOSTIC CAPABILITIES

REQUEST FOR CERTIFICATE OF CORRECTION
PURSUANT TO 37 CFR 1.323 AND 1.322

Attention: Certificate of Correction Branch
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Upon reviewing the above-identified patent, Patentee noted typographical errors which should be corrected. A listing of the errors to be corrected is attached.

The typographical errors marked with an "A" on the attached list are found in the application as filed by applicant. Payment in the amount of $100.00 covering the fee set forth in 1.20(a) is enclosed.

The typographical errors marked with a "P" on the attached list are not in the application as filed by applicant. Also given on the attached list are the documents from the file history of the subject patent where the correct data can be found.

The errors now sought to be corrected are inadvertent typographical errors the correction of which does not involve new matter or require reexamination.

Transmitted herewith is a proposed Certificate of Correction effecting such corrections. Patentee respectfully solicits the granting of the requested Certificate of Correction.

The Commissioner is authorized to charge any deficiency of up to $300.00 or credit any excess in this fee to Deposit Account No. 04-0100.

Dated: February 21, 2008

Respectfully submitted,

By _____

Flynn Barrison
Registration No.: 53,970
DARBY & DARBY P.C.
P.O. Box 770
Church Street Station
New York, New York 10008-0770
(212) 527-7700
(212) 527-7701 (Fax)
Attorneys/Agents For Applicant

PTO/SB/44 (04-05)
Approved for use through 04/30/2007. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,328,376 Page 1 of 1

APPLICATION NO.: 10/698,989

ISSUE DATE : February 5, 2008

INVENTOR(S) : McGuire et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the face page, in field (56), under "Other Publications", in column 2, line 1, delete "Wikipeida's" and insert - - Wikipedia's - -, therefor.

On the face page, in filed (56), under "Other Publications", in column 2, line 5, delete "20003" and insert - - 2003 - -, therefor.

In column 3, line 53, delete "programattically" and insert - - programatically - -, therefor.

In column 17, line 37, in Claim 19, after "the" delete "a".

MAILING ADDRESS OF SENDER (Please do not use customer number below):
John W. Branch, Esq.
DARBY & DARBY P.C.
P.O. Box 770
Church Street Station
New York, New York 10008-0770

Darby & Darby

Issued Patent Proofing Form File#: **20910/0206207-US0**

Note: **P** = PTO Error **A** = Applicant Error

US Serial No.: **10/698,989** US Patent No.: **US 7,328,376 B2** Issue Dt.: **Feb. 5, 2008**

Title: **ERROR REPORTING TO DIAGNOSTIC ENGINES BASED ON THEIR DIAGNOSTIC CAPABILITIES**

Sr. No.	P/A	Original		Issued Patent		Description Of Error
		Page	Line	Column	Line	
1	P	Page 1 of 1 List of references cited by examiner (09/06/2006)	Entry 1 Line 1 (Non-Patent Documents)	First Page Col. 2 (Other Publications)	1	Delete "Wikipeida's" and insert - - Wikipedia's - -, therefor.
2	P	Page 1 of 1 List of references cited by examiner (05/30/2007)	Entry 1 Line 2 (Non-Patent Documents)	First Page Col. 2 (Other Publications)	5	Delete "20003" and insert - - 2003 - -, therefor.
3	A	Page 5 Specification (10/31/2003)	22	3	53	Delete "programattically" and insert - - programatically - -, therefor.
4	A	Page 5 Claims (08/28/2007)	Claim 20 Line 1	17	37	In Claim 19, after "the" delete "a".

US007328376B2

(12) **United States Patent**
McGuire et al.

(10) **Patent No.: US 7,328,376 B2**
(45) **Date of Patent: Feb. 5, 2008**

(54) **ERROR REPORTING TO DIAGNOSTIC ENGINES BASED ON THEIR DIAGNOSTIC CAPABILITIES**

(75) Inventors: **Cynthia A. McGuire**, San Jose, CA (US); **Timothy P. Haley**, Boulder, CO (US); **Andrew M. Rudoff**, Boulder, CO (US); **Michael W. Shapiro**, San Francisco, CA (US); **Matthew T. Simmons**, San Francisco, CA (US)

(73) Assignee: **Sun Microsystems, Inc.**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 524 days.

(21) Appl. No.: **10/698,989**

(22) Filed: **Oct. 31, 2003**

(65) **Prior Publication Data**

US 2005/0102567 A1 May 12, 2005

(51) **Int. Cl.**
G06F 11/00 (2006.01)
G06F 11/07 (2006.01)

(52) **U.S. Cl.** **714/48**; 714/57; 714/26; 714/25

(58) **Field of Classification Search** 714/25, 714/26, 48, 57
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,922,491	A *	5/1990	Coale	714/26
6,260,059	B1 *	7/2001	Ueno et al.	709/202
6,343,236	B1 *	1/2002	Gibson et al.	700/79
6,487,677	B1 *	11/2002	Jantz et al.	714/2
6,732,124	B1 *	5/2004	Koseki et al.	707/202
6,742,141	B1 *	5/2004	Miller	714/26
7,082,384	B2 *	7/2006	Sasaki et al.	702/184
2001/0013109	A1 *	8/2001	Daimon et al.	714/57
2004/0059966	A1 *	3/2004	Chan et al.	714/48

FOREIGN PATENT DOCUMENTS

EP	1109101 A2	6/2001
WO	WO02/041105 A2	5/2002
WO	WO02/041105 A3	5/2002

OTHER PUBLICATIONS

Wikipeida's Cache Article revision from Oct. 27, 2003 http://en.wikipedia.org/w/index.php?title=Cache&oldie=1754687&printable=yes.*

The Many Faces of Publish/Subscribe by Eugster, Felber, Guerraoui, and Kermarrec Published Jun. 20003 by ACM Press in ACM Computing Surveys vol. 35 Issue 2, pp. 114-131 ISSN: 0360-0300.*

International Search Report (EP) mailed on Jan. 16, 2006.

* cited by examiner

Primary Examiner—Gabriel Chu
Assistant Examiner—Joseph Schell
(74) *Attorney, Agent, or Firm*—Darby & Darby PC; John W. Branch

(57) **ABSTRACT**

A method, apparatus, and computer program product diagnosing and resolving faults is disclosed. A disclosed fault management architecture includes a fault manager suitable having diagnostic engines and fault correction agents. The diagnostic engines receive error information and identify associated fault possibilities. The fault possibility information is passed to fault correction agents, which diagnose and resolve the associated faults. The architecture uses logs to track the status of error information, the status of fault management exercises, and the fault status of system resources. Additionally, a soft error rate discriminator can be employed to track and resolve soft (correctible) errors in the system. The architecture is extensible allowing additional diagnostic engines and agents to be plugged in to the architecture without interrupting the normal operational flow of the computer system.

24 Claims, 5 Drawing Sheets

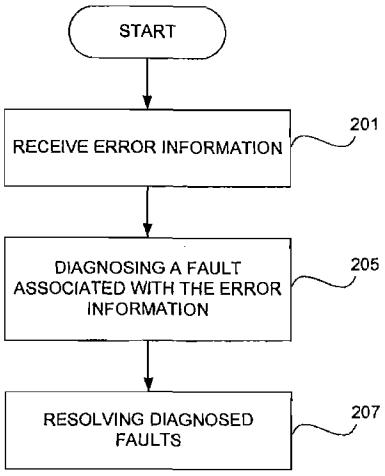

FIG. **2** is a flow diagram illustrating one simplified exemplary method embodiment of the present invention.

FIG. **3** is a flow diagram illustrating one exemplary method embodiment of a fault management exercise of the present invention.

FIGS. **4**A and **4**B illustrate an embodiment of a computer system suitable for implementing embodiments of the present invention.

It is to be understood that, in the drawings, like reference numerals designate like structural elements. Also, it is understood that the depictions in the Figures are not necessarily to scale.

DETAILED DESCRIPTION OF THE DRAWINGS

The present invention has been particularly shown and described with respect to embodiments and specific features thereof. The embodiments set forth herein below are to be taken as illustrative rather than limiting. It should be readily apparent to those of ordinary skill in the art that various changes and modifications in form and detail may be made without departing from the spirit and scope of the invention.

When a computer system encounters a system interrupt (an error) the computer system can begin to function erratically or fail completely. A computer system error is a *symptom* detected by the computer system in response to a fault (i.e., the underlying problem in the system that caused the error). Typical examples of such errors include commands that time out, bus errors, I/O errors, ECC memory (Error-Correcting Code memory) errors, unexpected software results, and the like. Other errors include the typical 256 software interrupts that are commonly found on interrupt vector tables. Such software interrupts are commonly referred to as traps or exceptions. Other error examples include hardware interrupts (e.g., IRQ line failures etc.). The faults that cause such errors are legion. A few common examples include device failures, bus line failures, disconnected cables, memory failures, and many, many more. It is important that faults causing these errors be identified and corrected as soon as possible to enable efficient system operation.

The embodiments of the present invention go beyond current approaches to fault diagnosis and correction and do not require extensive manual action on the part of the system administrator. The embodiments go beyond approaches that are limited to general error reporting and rudimentary guidance as to which diagnostic tools may be useful in finding the responsible fault. The embodiments of the invention do not always require the system administrator to evaluate errors to determine which diagnostic tools to use next and then acquire further error information in order to diagnose the nature of the fault. The embodiments of the invention can programattically take action to correct faults. The systems and method embodiments of the invention can operate with the system "on-line". This goes beyond existing approaches have no ability to capture data, diagnose faults, and correct faults "on the fly" (while the system is online operating normally). Additionally, embodiments of the present invention are readily extensible. Thus, when new diagnostic tools become available, they can simply be plugged into the system and used. There is no need for the system to be taken offline and no need for the entire messaging sub-scheme to be reconfigured in order to patch in the new tool as is the case with conventional approaches.

The following detailed description describes various method and apparatus embodiments of a fault management architecture used in a computer system. In general, a fault management architecture constructed in accordance with the principles of the invention operates in an automated manner to collect error information, evaluate the error information, and diagnose faults associated with the error information. Additionally, the fault management architecture takes action to resolve the faults. Such resolution can also be automated. Embodiments of the fault management architecture operate at the user level of the operating system (O/S) and not at the kernel level and so do not require that the system be taken offline in order to operate. In fact, the fault management architecture of the present invention can be continuously operating whenever the system is operating. Moreover, the fault management architecture can be readily updated with improved features without taking the system offline. For example, new or updated diagnostic engines **102** and fault correction agents **103** can be added to (or removed from) the system while the computer system is operating without interfering with the normal operation of the computer system.

For purposes of this disclosure there is a user level and a kernel level. System and end-user application software runs at the "user-level". Additionally, there is a kernel level. As is known to those having ordinary skill in the art, the kernel is a special program that manages system resources (e.g., software and hardware). The kernel insulates applications from system hardware while providing them with controlled access to hardware and essential system services including, but not limited to I/O management, virtual memory, and scheduling.

FIG. **1** depicts one example of a suitable fault management architecture constructed in accordance with the principles of the invention. In the depicted embodiment, the fault management architecture **100** operates in a computer system at the user level. The advantage of operating at the user level means that the operation of the fault management architecture does not interfere with the operation of the kernel. Thus, the computer system can operate effectively at the same time the fault management architecture is operating. The fault management architecture includes a fault manager **101**, which includes a plurality of diagnostic engines **102** (e.g., DE_1, DE_2, . . . DE_n) and a plurality of fault correction agents **103** (e.g., A_1, A_2, . . . A_m). The fault manager **101** can optionally include a soft error rate discriminator (SERD) **105** whose function and utility will be explained in greater detail hereinbelow. The fault management architecture **100** also includes a data capture engine **110**. In some embodiments, the data capture engine **110** can optionally be included as part of the fault manager **101** itself. Another advantage of operating the fault management architecture at the user level is that the diagnostic engines **102** and the fault correction agents **103** can be plugged into (or unplugged from) the computer system without interfering normal system operation. The process of capturing data through fault diagnosis and resolution is referred to as a fault management exercise. Processes and methods for facilitating such fault management exercises are described in greater detail elsewhere herein.

Referring again to FIG. **1**, the data capture engine **110** is a set of computer readable program instructions for receiving and processing error information from the computer system. For example, the data capture engine **110** can capture error information in many different software components (and resources) including, but not limited to, a kernel module, device drivers, trap handlers, interrupt handlers, and user-level applications. The data capture engine **110** passes this error information to the fault manager **101** for further processing. The data capture engine **110** operates

least one of: an analysis of at least one of computer resource failure history, system management policy, and relative probability of occurrence for each fault possibility.

15. The fault management architecture of claim **1** wherein the fault manager stores provided error reports in a log comprising an error report log and wherein the error report log tracks the status of the provided error reports.

16. The fault management architecture of claim **1** wherein the fault manager includes a soft error rate discriminator that:

receives error information concerning correctable errors;

wherein the soft error rate discriminator is configured so that when the number and frequency of correctable errors exceeds a predetermined threshold number of correctable errors over a predetermined threshold amount of time, these errors are deemed recurrent correctable errors that are sent to the diagnostic engines for further analysis;

wherein the diagnostic engine receives a recurrent correctible error message and

diagnoses a set of fault possibilities associated with the recurrent correctible error message; and

wherein a fault correction agent receives the set of fault possibilities from the diagnostic engines and then resolves the diagnosed fault.

17. The fault management architecture of claim **16** wherein the soft error rate discriminator receives error information concerning correctable errors from the diagnostic engine.

18. The fault management architecture of claim **16** wherein the diagnostic engine that identifies a set of fault possibilities associated with the recurrent correctable error message further determines associated probabilities of occurrence for the set of fault possibilities associated with the recurrent correctable error message.

19. The fault management architecture of claim **18** wherein the a fault correction agent receives the set of fault possibilities and associated probabilities of occurrence from the diagnostic engines and the agent then takes appropriate action to resolve the set of fault possibilities.

20. The fault management architecture of claim **1** wherein the fault manager includes a soft error rate discriminator that:

receives error information concerning soft errors;

wherein the soft error rate discriminator is configured so that when the number and frequency of soft errors exceeds a predetermined threshold number of soft errors over a predetermined threshold amount of time, these soft errors are deemed recurrent soft errors that are sent to the diagnostic engines for further analysis;

wherein the diagnostic engine receives a recurrent soft error message and diagnoses a set of fault possibilities associated with the recurrent correctable error message; and

wherein a fault correction agent receives the set of fault possibilities from the diagnostic engines and then resolves the diagnosed fault.

21. The fault management architecture of claim **1** further including a fault management administrative tool that is configured to enable a user to access the logs to determine the fault status and error history of resources in the computer system.

22. The fault management architecture of claim **1** further including a fault management statistical file that can be reviewed to determine the effectiveness of the diagnostic engines and fault correction agents at diagnosing faults and resolving faults.

23. The fault management architecture of claim **1** wherein the computer system comprises a single computer device.

24. The fault management architecture of claim **1** wherein the computer system comprises a plurality of computers forming a network.

* * * * *